

Contents

| | |
|---|----------|
| 1 Routine/Function Prologues | 2 |
| 1.0.1 glbprecip_cmap (Source File: obsprecip.F90) | 2 |
| 1.0.2 glbprecip_pers (Source File: obsprecip.F90) | 3 |
| 1.0.3 glbprecip_pers (Source File: obsprecip.F90) | 4 |

1 Routine/Function Prologues

1.0.1 glbprecip_cmap (Source File: *obsprecip.F90*)

Includes reading routines for global CMAP precipitation product Used instead of GDAS/GEOS precipitation forcing

REVISION HISTORY:

```
17 Jul 2001: Jon Gottschalck; Initial code
04 Feb 2002: Jon Gottschalck; Added necessary code to use global precip
              observations with domain 3 (2x2.5)
30 Jul 2002: Jon Gottschalck; Added code to use Huffman and Persiann precip data
```

INTERFACE:

```
subroutine glbprecip_cmap( ld, gindex, fname, ferror_cmap, filehr)
```

USES:

```
use ldas_module
use obsprecipforcing_module, only: obsprecip
implicit none
```

ARGUMENTS:

```
type (ldasdec) ld
integer :: gindex(ld%d%lnc, ld%d%lnr)
character(len=80) :: fname           ! Filename variable for datafile
integer :: ferror_cmap
integer :: filehr
```

CONTENTS:

```
allocate (precip_regrid(ld%d%lnc,ld%d%lnr))
precip_regrid = -1.0
!-----
! Set necessary parameters for call to interp_gdas
!-----
ism      = 0
udef     = ld%d%udef
jj       = 0
if (mod((filehr),12).eq.0) then
  lugb=25
else
  lugb=26
endif
lugi    = 0
jpds    = -1
jpds(5) = 59
jpds(6) = 1
jpds(7) = 0
```

```

jgds      = 0
call baopen (lubb, fname, iret)
if (iret == 0 ) then
call getgb (lubb, lugi, ncmap, jj, jpds, jgds, kf, k, kpds, kgdscmap, lb, cmapin, iret)
call interp_cmap(kpds, kgdscmap, ncmap, cmapin, lb, ld%d%kgds, &
ld%d%lnc, ld%d%lnr, precip_regrid)

do j = 1,ld%d%lnr
do i = 1,ld%d%lnc
    if(precip_regrid(i,j) .eq. -1) then
        print*, j,i,precip_regrid(i,j)
    endif
    if (precip_regrid(i,j) .ne. -1.0) then
        index = gindex(i,j)
        if(index .ne. -1) then
            obsprecip(index) = precip_regrid(i,j)*3600.0
        endif
    endif
enddo
enddo

call baclose (lubb, jret)

ferror_cmap = 1
close(10)
print*, "Obtained CMAP CPC precipitation data ", fname
else
    print*, "Missing CMAP CPC precipitation data ", fname
    ferror_cmap = 0
endif
deallocate (precip_regrid)

```

1.0.2 glbprecip_pers (Source File: obsprecip.F90)

Includes reading routines for global PERSIANN precipitation product Used instead of GDAS/GEOS precipitation forcing

INTERFACE:

```
subroutine glbprecip_pers ( ld, gindex, name_pers, ferror_pers )
```

USES:

```
use ldas_module
use obsprecipforcing_module, only: obsprecip
```

CONTENTS:

```

fname = name_pers
obsprecip = -1.0
!-----
! Determine offset in number of rows from 60 S
! since PERSIANN starts at 50 S
!-----
open(unit=10,file=fname, status='old',access='direct', &
      form='unformatted',recl=xd*yd*4,iostat=ios)

if (ios .eq. 0) then
  read (10,rec=1) precip

  do i = 1,yd
    do j = 1,xd
      index = gindex(j,i)
      col(index) = j
      row(index) = i
      if (index .ne. -1) then
        obsprecip(index) = precip(j,i)
      endif
    enddo
  enddo
  ferror_pers = 1
  close(10)
  print*, "Obtained PERSIANN precipitation data ", fname
else
  print*, "Missing PERSIANN precipitation data ", fname
  ferror_pers = 0
endif

```

1.0.3 glbprecip_pers (Source File: obsprecip.F90)

Includes reading routines for global HUFFMAN precipitation product Used instead of GDAS/GEOS precipitation forcing

INTERFACE:

```
subroutine glbprecip_huff ( ld, gindex, name_huff, ferror_huff )
```

USES:

```
use ldas_module
use obsprecipforcing_module, only: obsprecip
```

CONTENTS:

```

fname = name_huff
!-----
```

```

! Fill necessary arrays to assure not using old HUFFMAN data
!-----
precip = -1.0
obsprecip = -1.0
!-----
! Find HUFFMAN precip data, read it in and assign to forcing precip array.
! Must reverse grid in latitude dimension to be consistent with LDAS grid
!-----
open(unit=10,file= fname, status='old', &
&           access='direct',recl=xd*yd*4, &
&           form='unformatted',iostat=ios)

if (ios .eq. 0) then
  read (10,rec=1) head (1:2880),&
  ( ( rr (i, j), i = 1, xd ), j = 1, yd)
do i = 1,yd
  do j = 1,xd
    if (j .lt. 721) then
      precip(j,i) = float(rr(j+720,yd+1-i)) / 100.0
      if ( rr(j+720,yd+1-i) .eq. ibad ) precip(j,i) = -1.0
      if ( rr(j+720,yd+1-i) .lt. 0.0 ) precip(j,i) = -1.0
    else
      precip(j,i) = float(rr(j-720,yd+1-i)) / 100.0
      if ( rr(j-720,yd+1-i) .eq. ibad ) precip(j,i) = -1.0
      if ( rr(j-720,yd+1-i) .lt. 0.0 ) precip(j,i) = -1.0
    endif
  enddo
enddo
!-----
! Interpolating to desired domain and resolution
! Global precip datasets not used currently to force NLDAS
!-----
do i = 1,yd
  do j = 1,xd
    index = gindex(j,i)
    if (index .ne. -1) then
      obsprecip(index) = precip(j,i)
    endif
  enddo
enddo

ferror_huff = 1
close(10)
print*, "Obtained HUFFMAN precipitation data ", fname
else
  print*, "Missing HUFFMAN precipitation data ", fname
  ferror_huff = 0
endif

```

